基于 LDA 和 word2vec 的英文作文跑题检测 *

曲　强，崔荣一，赵亚慧†

(延边大学 计算机科学与技术学科 智能信息处理研究室，吉林 延吉 133002)

摘　要：针对目前国内的英语作文辅助批阅系统缺少准确而高效的跑题检测算法的问题，提出了一种结合 LDA 和 word2vec 的跑题检测算法。该算法利用 LDA 模型对文档建模并通过 word2vec 对文档训练，利用得到的文档主题和词语之间的语义关系，对文档中各主题及其特征词计算概率加权和，最终通过设定合理阈值筛选出跑题作文。实验中通过改变文档的主题数而得到的不同 F 值，确定了最佳主题数。实验结果表明新方法比基于向量空间模型的方法更具有效性，可以检测到更多的跑题作文并且准确率较高，F 值达到 89%以上，实现了作文跑题检测的智能化处理，可以有效地应用在英语作文教学中。

关键词：作文跑题检测；向量空间模型；潜在狄利克雷分配；词语间语义关系

中图分类号：TP391.1　　doi: 10.3969/j.issn.1001-3695.2017.08.0724

Off-topic detection for English essays based on LDA and word2vec

Qu Qiang, Cui Rongyi, Zhao Yahui†

(*Intelligent Information Processing Laboratory*, *Dept. of Computer Science & Technology*, *Yanbian University*, *Yanji Jilin* 133002, *China*)

Abstract: Aiming at the problem that the lack of accurate and efficient off-topic detection algorithm for the current English composition teaching system in China, proposed an off-topic detection algorithm of LDA and word2vec in this paper. The algorithm used LDA to model the documents and train it with word2vec, with obtained semantic relation between document's topic and words, calculated the probability weighted sum of each topic and its feature words in the document. Finally, by setting reasonable threshold, selected the off-topic essays. According to the different F values for the different number of topics in the document, determined the optimum number of topics in the experiment. The experimental results show that, compared to traditional vector space model, the proposed method can detect more off-topic essays with higher accuracy, and the F value is above 89%, which realizes the intelligent processing of off-topic essays detection, and may applies effectively in English essays teaching.

Key Words: off-topic essays detection; vector space model (VSM); latent Dirichlet allocation (LDA); semantic relations between words

0　引言

作文是一种表达情感和传递信息的重要手段，而主题则是作文的灵魂。一篇作文最重要的就是主题明确并且正确，否则容易造成混淆和误解，甚至跑题。作文跑题的原因很多，可能是作者有意为之，也可能是无意间的提交错误[1]。

作文跑题检测用于判断一篇作文是否跑题，其核心内容是计算文本之间的相似度[2]，文本相似度是表示文本间相似程度的衡量参数。目前最常用、最经典的文本表示模型是向量空间模型，基于向量空间模型的 TF-IDF 算法是使用最广泛的文本相似度计算的方法。这种方法以词在文档中出现频率以及在文档集中出现该词的频率来表征词的权重，通过计算向量之间的余弦值来计算文本的相似度。词袋模型方法虽然简单而且有一定效果，但是这种方法忽略了文档中词项本身的语义信息，没有考虑到词与词之间的语义相似度。比如对于英文单词“like”和“love”，它们都可以表示为喜欢的意思，但在向量空间模型中，就会把它们当作两个独立的词项。为了解决这个缺点，有研究人员提出了词扩展的方法，比如使用 WordNet、HowNet 等词典进行词扩展。文献[3]提出了基于 WordNet 词扩展计算英语词汇语义相似度的方法，文献[4]提出了基于 HowNet 计算词汇语义相似度的方法。这些方法都很依赖人工构造的词典，出现新词的时候可能会遇到很多问题。

基金项目：国家语委“十二五”科研规划 2015 年度科研项目（YB125-178）

作者简介：曲强（1992-），男，吉林长春人，硕士研究生，主要研究方向为自然语言处理；崔荣一（1962-），男，吉林延吉市人，教授，硕导，博士，主要研究方向为机器学习、自然语言处理；赵亚慧（1974-），女（通信作者），吉林长春人，副教授，硕导，硕士，主要研究方向为自然语言处理（yhzhao@ybu.edu.cn）.

本文是针对以上方法的不足，提出了一种新的文本相似度计算方法并根据此方法进行英文作文的跑题检测。该算法通过 LDA 主题模型对文档集建模，得到每个文档的主题和主题的特征词以及它们的概率分布，并和 word2vec 训练得到的词与词之间的语义关系进行结合，计算出文档的各个主题的概率加权和，判定作文是否偏离主题。该方法的提出可以有效地检测到跑题的作文，与传统的向量空间模型相比，本文的方法不但可以得到词项之间的更多的语义信息，还可以通过对文档建模得到文档的主题分布情况，弥补了传统向量空间模型方法没考虑到词本身语义信息的不足。

1 LDA 建模

1.1 LDA 模型

LDA（latent dirichlet allocation）模型是由 Blei 等人提出的一个“文本—主题—词”的三层贝叶斯产生式模型[5]，它是在概率隐性语义索引(probabilistic latent semantic analysis, pLSA)上扩展得到的三层贝叶斯概率模型，该模型包含词、主题和文档三层结构。该模型是一种非监督的机器学习算法，可以用来识别大规模文档集或语料库中潜在的主题信息。它采用了词袋模型（bag of words）的方法，这种方法将每一篇文档视为一个词频向量，从而将文本信息转换为了便于建模计算的数字信息。该模型基于这样一种前提假设：文档是由若干个隐含主题构成，而这些主题是由文本中若干个特定词汇构成，忽略文档中的句法结构和词语出现的先后顺序[6]。

LDA 主题模型可以用一个概率图模型表示，其表示形式如图 1 所示。

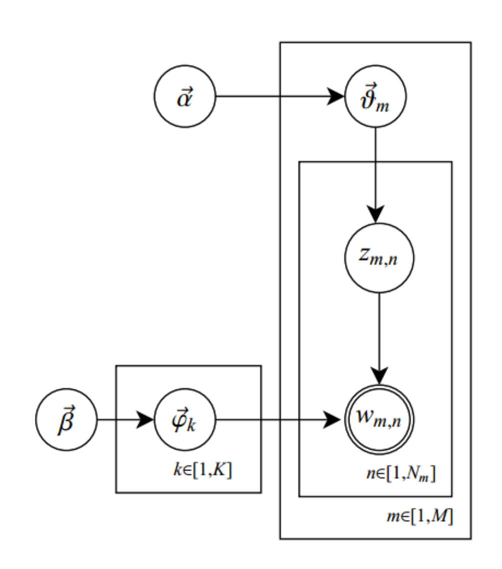


图 1 LDA 模型图

LDA 模型由超参数 α 和 β 确定，其中 α 表示文档集合中隐含主题之间的相对强弱，β 反映了所有隐含主题的自身的概率分布。在图 1 中 M 表示文档集的文档数，K 表示文档集中的主题数，N 表示每篇文档包含的特征词数，θ_m 表示第 m 篇文档中所有主题的概率分布，ϕ_k 表示某一特定主题下的特征词的概率分布。

1.2 Gibbs 抽样

在构建 LDA 模型的过程中需要进行模型参数的估计，比较常用的估计方法主要有变分贝叶斯推理、期望传播算法和 Collapsed Gibbs 抽样等，基于 Gibbs 抽样的参数推理方法容易理解而且实现简单，能够非常有效地从大规模文本集中抽取主题[7]。因此 Gibbs 抽样算法成为了当前最流行的 LDA 模型抽取算法。

Gibbs 抽样方法是一个简单的并且应用广泛的 MCMC(Markov chain Monte Carlo)算法，Griffiths 提出将 Gibbs 采样方法应用于 LDA 模型的参数估计[8]。每个主题下的特征词项概率分布和每篇文档的主题概率分布是在 LDA 模型中最重要的两个参数。

Gibbs 抽样算法具体步骤如下（该算法具体推导过程可以详见文献[9]）：

a)初始化。主题 z_i 被初始化为 1 到 T 之间的某个随机整数，i 从 1 循环到 N，N 是语料库中所有出现在文本中的特定词的个数，此为 Markov 链的初始状态。

b)循环采样。经过迭代足够多的次数以后，直到 Markov 链接近目标分布，此时的主题 z_i 可以按照如下公式估算 ϕ 和 θ 的值。

$$\hat{\phi}_k^{(t)} = \frac{n_k^{(t)} + \beta_t}{\sum_{t=1}^{V} n_k^{(t)} + \beta_t} \tag{1}$$

$$\hat{\theta}_m^{(k)} = \frac{n_m^{(k)} + \alpha_k}{\sum_{k=1}^{K} n_m^{(k)} + \alpha_k} \tag{2}$$

其中：$n_k^{(t)}$ 表示的是第 k 个主题出现第 t 个特征词的次数，$n_m^{(k)}$ 表示的是第 m 篇文档出现第 k 个主题的次数。通过 Gibbs 抽样间接得到的 ϕ 和 θ 值，记为后验概率 $P(z_i = k \mid z_{-i}, w)$，其计算公式如下：

$$P(z_i = k \mid z_{-i}, w) \propto \frac{n_{m,\neg i}^{(k)} + \alpha_t}{\sum_{k=1}^{K}(n_{m,\neg i}^{(k)} + \alpha_t)} * \frac{n_{k,\neg i}^{(t)} + \beta_k}{\sum_{t=1}^{V}(n_{k,\neg i}^{(t)} + \beta_k)} \tag{3}$$

式（3）中因为 z_i 表示第 i 个词项对应的主题变量，$\neg i$ 表示不包括其中的第 i 项，所以 $z_{\neg i}$ 表示所有主题 z_k $(k \neq i)$ 的概率分配。$z_{k,\neg i}^{(t)}$ 表示特征词 t 属于主题 k 的词频；$z_{m,\neg i}^{(k)}$ 表示文档 m 分配给主题 k 的特征词集的规模。

1.3 LDA 建模过程

本文在对进行 LDA 建模之前，对于给定的文档集合 $D=\{d_1, d_2, \cdots d_M\}$，需要对每篇文档 $d_m(d_m \in D)$ 进行预处理，主要包括分词、去停用词、去标点符号等操作，将处理后的每个词项用空格分隔保存，整理后获得对应的语料集，将其作为下一步的处理数据。

将处理后的语料以一篇文档的形式呈现出来，构建出文档-词项矩阵。最终文本表示形式如式（4）所示。

$$D=\begin{bmatrix} w_{11}, & w_{12}, & \cdots, & w_{1n} \\ \vdots & \vdots & & \vdots \\ w_{m1}, & w_{m2}, & \cdots, & w_{mn} \\ \vdots & \vdots & & \vdots \\ w_{M1}, & w_{M2}, & \cdots, & w_{Mn} \end{bmatrix} \tag{4}$$

其中：M 代表文档总数，m 代表文档序号，w_{mn} 表示第 m 篇文档的第 n 个词项。

对于语料库中的每篇文档，LDA 给出了如下的生成过程：

a）从狄利克雷分布 α 中取样生成第 m 个文档的主题分布 θ_m；

b）从主题的多项式分布 θ_m 中取样生成第 m 个文档的第 n 个词的主题 $z_{m,n}$；

c）从狄利克雷分布 β 中取样生成主题 $z_{m,n}$ 对应的词语分布 $\phi_{z_{m,n}}$；

d）从词语的多项式分布 $\phi_{z_{m,n}}$ 中采样最终生成词语 $w_{m,n}$。

由于 LDA 模型认为一篇文章是有多个主题的，而每个主题又对应着不同的词。一篇文章的构造过程，首先是以一定的概率选择某个主题，然后再在这个主题下以一定的概率选出某一个词，这样就生成了这篇文章的第一个词。不断重复这个过程，就生成了整片文章。当然这里假定词与词之间是没顺序的。

本文参数估计利用 MCMC[10]方法中的 Gibbs 抽样[11]算法，它可以看做是文档生成过程的逆过程，即在已知文档集（文档生成的结果）的情况下，通过参数估计得到参数值。根据图 1 的模型图，可以得到一篇文档的概率分布：

$$p(\omega\,|\,\alpha,\beta)=\int p(\theta\,|\,\alpha)(\prod_{n=1}^{N}\sum_{z_n}p(z_n\,|\,\theta)p(\omega\,|\,z_n,\beta))d\theta \tag{5}$$

通过 Gibbs 抽样算法可以基于语料训练 LDA 模型，训练的过程就是通过 Gibbs 抽样得到文档集中的主题和特征词的样本，算法收敛后得到的最终样本便可以对模型进行参数估计。

通过上述步骤和分析，针对本文实验的需求，对式（4）得到的文档-词项矩阵，使用 LDA 模型对预处理后的文档集 D 进行建模，从而得到文档 d_m 的主题 t_i 和其主题概率分布 $P(t_i\,|\,d_m)$，其中 $t_i \in T, T=\{t_1,t_2,\cdots,t_K\}$，并得到主题 t_i 的特征词 w_n 及其特征词概率分布 $P(w_n\,|\,t_i)$，其中 $w_n \in W, W=\{w_1,w_2,\cdots,w_N\}$。

2　基于 LDA 和 word2vec 的主题相关度计算

LDA 模型对文档的表示是用概率的形式对主题和主题对应的特征词进行抽取，有一定的不确定性，为了更精确地表达文档中词项的语义信息，本文引入 word2vec 方法更好的表达词与词之间的语义信息。通过该方法，与 LDA 建模后主题的特征词进行计算词项之间的相似度，最后得到主题相关度。

2.1　word2vec

近几年，随着深度学习的迅速发展，基于神经网络的自特征抽取的词向量表示方法越来越受到广大研究者的关注。Mikolov 等人通过借鉴 Bengio 提出的 NNLM(Neural Network Language Model)模型以及 Hinton 的 Log_Linear 模型，提出了 word2vec 语言模型[12]，用于计算词向量。Google 公司在 2013 年开放了 word2vec 这一款用于训练词向量的开源软件工具[13]。word2vec[14]模型可以根据给定的语料库，通过优化后的训练模型快速有效地将一个词语表达成实数值的向量形式，它可以通过利用词的上下文信息把对文本内容的处理简化为 K 维向量运算，而向量空间上的相似度可以用来表示文本语义上的相似度。word2vec 输出的词向量可以用来做很多 NLP 相关的工作，比如情感分类、找近义词、词性分析等。而 word2vec 另一个特点就是高效性，Mikolov 等[15]指出一个优化的单机版本一天可训练上千亿个词。其为自然语言处理领域的应用研究提供了新的工具。

word2vec 包含了两种训练模型，采用的架构模型分别是 CBOW(Continuous Bag-Of-Words)模型和 Skip-Gram 模型。其原理示意图如图 2 所示。

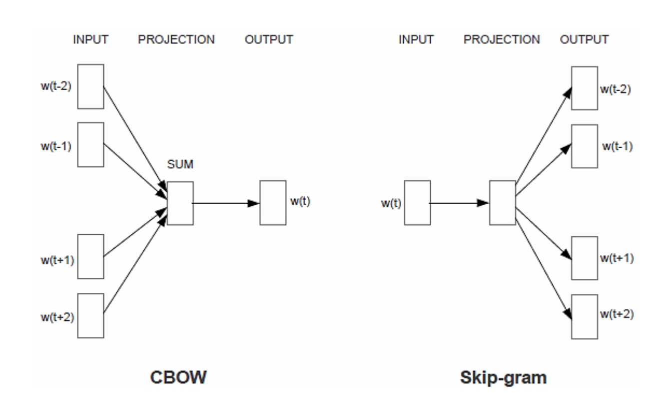


图 2　CBOW 模型和 Skip-gram 模型原理示意图

从图 2 可以明显地看出，CBOW 和 Skip-gram 模型均包含输入层、投影层和输出层。其中，CBOW 模型通过上下文来预测当前词的词向量，即将当前词上下文对应的连续词语表示成词袋的形式，将训练的目标向量选为上下文词向量的求和。而 Skip-gram 模型生成词向量的方式恰好与 CBOW 模型相反，它仅通过当前词来预测其上下文。通过这两个模型，word2vec 就能够很全面地考虑上下文信息，因此，可以能取得较好的效果。

2.2　主题相关度计算

在对文档进行主题相关度计算之前，需要通过 word2vec 对文档集进行训练，从而得到词项之间的语义信息。

对于本文的英文语料而言，word2vec 可以根据词语之间的空格来识别不同的词语。经过 word2vec 训练之后，能够得到每个词语的向量表示，计算两个向量的余弦值来表示两个词语的语义相似度距离，余弦值越大，表示两个词语的语义越相近。例如两个 n 维向量 $a(x_{11},x_{12},\cdots x_{1n})$ 和 $b(x_{21},x_{22},\cdots x_{2n})$，余弦值的计算公式如下：

$$\cos(a,b)=\frac{\sum_{k=1}^{n}x_{1k}x_{2k}}{\sqrt{\sum_{k=1}^{n}x_{1k}^2}\sqrt{\sum_{k=1}^{n}x_{2k}^2}} \tag{6}$$

将经过训练后得到的词向量表示信息存储到文件中，便于后续步骤计算词向量的相似度使用。

根据上面得到的信息，对每篇文档的每个词项 w_j，利用 word2vec 计算该词项和在 t_i 主题下的特征词 w_n 的余弦相似度 $\cos(w_j,w_n)$。则词项 w_j 和主题 t_i 的相关度为 w_j 和在 t_i 下的各个特征词的余弦相似度的概率加权和 $S(w_j,t_i)$，可以用如下公式表示：

$$S(w_j,t_i)=\sum_{n=1}^{N}P(w_n\mid t_i)\times\cos(w_j,w_n) \tag{7}$$

所以可以得到词项 w_j 和文档 d_m 的相关度，即 w_j 和 d_m 的各个主题的相关度的概率加权和 $S(w_j,d_m)$，用公式表示为

$$S(w_j,d_m)=\sum_{i=1}^{K}P(t_i\mid d_m)\times S(w_j,t_i) \tag{8}$$

最后把文档的每个词项得到的 $S(w_j,d_m)$ 值加和。公式表示如下：

$$S_m=\sum_{j=1}^{J}S(w_j,d_m) \tag{9}$$

3 跑题检测算法

跑题检测算法首先对文档集合进行预处理，通过预处理后建立文档—词项矩阵，接着通过 LDA 模型对文档集建模，得到文档的主题及其分布，和主题下的特征词及其分布。再用 word2vec 对文档集进行训练并保存训练的结果，然后把 LDA 和 word2vec 得到的信息进行组合。最后根据本文设置选取的阈值来对每篇文档进行筛选，从而找出跑题的文档。

跑题检测算法不但可以通过 LDA 得到文档的主题信息，还可以通过 word2vec 训练的词向量得到更准确的词项包含的语义信息。以上是对于作文跑题检测有很好的效果的重要因素。

跑题检测算法的具体步骤设计如下：

a)首先对文档集合进行预处理操作。对于英文文档的预处理，需要对文档中的内容按空格进行分词、将每句中的首个单词和专有名词等大写字母和单词统一转换为小写、去掉 the, a, an 等停用词、去掉所有标点符号、提取每个单词的词干（去掉单词的复数、-ing、-ed 等形式的词缀）等操作。如句子“we all like the book, it is so interesting.”，经过预处理后，结果为“like book interest”。

b)对预处理后的文档集合建立文档-词项矩阵。文档向量化后的表示结果形如式（4）所示，其中，矩阵中的第 i 行表示为第 i 篇文档，第 i 行的列数表示为该文档中包含词项的个数，第 i 行的第 j 列对应第 i 篇文档中的第 j 个词项。

c)进行 LDA 建模。对上述步骤建好的文档-词项矩阵中的每篇文档进行建模，由式（1）（2）分别得到第 m 篇文档的主题概率分布 θ_m 和第 k 个主题下的特征词的概率分布 ϕ_k 的值，根据概率值从大到小排序，从而得到每篇文档的主题及其概率分布和特征词及其概率分布。例如一篇英文文档主题概率分布的60%在讨论教育，40%是关于孩子，则在教育主题下，会出现“school、“students”、“education”等特征词项；在孩子主题下，特征词项有“children”、“women”、“family”等。

d)用 word2vec 训练词向量。以预处理后的文档集作为输入，用 word2vec 进行训练，输出为每个词对应的词向量。利用生成的词向量，通过式（6）计算和指定词语之间的距离（相似度）。比如指定词语为“woman”，将显示训练后的文本中与“woman”最接近的词语“man”以及它们之间的余弦距离为 0.685。训练后可以表达文档中词项之间的语义信息，变成向量信息并保存。

e)用 LDA 和 word2vec 对文档进行主题相关度计算。对每篇文档的每个词项用 word2vec 计算其与 LDA 建模后的第 i 个主题下的各个特征词的余弦相似度，利用式（7）计算各个特征词的概率加权和，然后按照式（8）对各个主题的概率加权和进行计算，最后根据式（9）把每个词项得到的主题相关度进行加和确定总相关度，并根据阈值筛选出跑题的作文。

算法中的 LDA 模型对文档集建模，利用 Gibbs 进行抽样，间接得到模型参数。通过参数估计可以得到文档中不同主题及其概率分布和不同主题的特征词及其概率分布，具有坚实的统计学基础。算法为了更精确的表示文档中的语义信息，加入 word2vec 来训练词向量的方法。该方法采用低维空间表示法，不但解决了维数灾难的问题，而且还挖掘了词与词之间的关联属性，从而提高了文本语义上的准确度。综上所述，算法结合了 LDA 和 word2vec 的各自优点，经过 word2vec 训练的结果使文档中词语间的语义关系表达的更准确，使得 LDA 建模后可以有效地判断文档本身的主题是否更切题，在低维的语义空间中得到了文档的主题相关度，通过相关度可以检测出跑题的文档。

4 实验结果及对比分析

本文实验收集了 6 个不同题目的大学英文作文，每个题目 205 篇，一共 1230 篇文档。每篇作文都有人工做好标注的打分结果，每个题目下的作文都有一定数量的跑题作文，满分 15 分的作文如果人工标注打分结果为 5 分以下本文就认为该作文是跑题的。实验结果检测到的跑题文档是为了和人工标注的打分结果中的跑题文档进行对比，从准确率、查全率和 F 值综合评价分析，进而验证实验中算法的有效性与实用性。

其中准确率是指正确检测出跑题的相关文档数与检测出跑题的文档总数的比例，用 P 来表示准确率，查全率是指正确检测出跑题的相关文档数与所有跑题的相关文档数的比例，用 R 表示查全率。假设用 T 来表示系统正确检测出的相关跑题文档数，用 A 来表示系统检测出的跑题文档总数，跑题相关文档的总数用 B 来表示，则准确率和查全率的计算公式如下：

$$P=\frac{T}{A}\times 100\% \tag{10}$$

$$R=\frac{T}{B}\times 100\% \tag{11}$$

从式（10）（11）的含义上得知，一般情况下准确率越高、查全率就越低，而查全率越高、准确率就越低。F 值可以调和它们互相牵制的影响，是一个兼顾准确率和查全率的综合指标，

其计算公式如下：

$$F = \frac{2PR}{P+R} \times 100\% \tag{12}$$

从式（12）可知，由于 F 值综合考虑了准确率和查全率的结果，当其较高时则说明算法比较理想。

实验中 LDA 模型使用了 Gibbs 抽样，在对文档主题建模的过程中，首先假定主题数目 K 为 2，本实验中超参数 α 取经验值[16]，$\alpha = 50/K$，它随着主题数目变化，超参数 β 也取固定的经验值[17]，$\beta = 0.01$，为了确保实验结果的准确性，Gibbs 抽样迭代次数设置为 1000 次。

在利用 word2vec 训练文档集的时候，因为 word2vec 提供了很多个超参数来调整训练过程，选择不同的参数对训练生成的词向量质量以及训练的速度都会有所影响，通过查阅文献[18]可以得知 word2vec 训练时的不同参数和每个参数所代表的含义，根据本实验的需求，用 word2vec 对文档集训练时的参数设置情况结果如表 1 所示。

表 1　word2vec 参数设置情况

超参数	参数说明	取值
size	词向量的维数	50
window	上下文窗口的大小	5
min-count	词语出现的最小阈值	1
cbow	是否使用 cbow 模型（0 为使用）	1

假定主题数 K 为 2 时，按照图 3 设计的算法，经过 LDA 对文档建模并和 word2vec 组合后，通过选取一定的阈值得到的跑题文档和人工标注的结果进行对比，根据式（10）~（12）得到相应的跑题检测的准确率、查全率和 F 值，最后计算出 6 个题目的平均结果。结果如表 2 所示。

表 2　主题数为 2 时的跑题检测结果

	题目 1	题目 2	题目 3	题目 4	题目 5	题目 6	平均值
准确率	94.74%	93.33%	93.75%	86.67%	61.54%	75%	84.17%
查全率	94.74%	100%	100%	86.67%	80%	75%	89.40%
F 值	94.74%	96.55%	96.77%	86.67%	69.57%	75%	86.55%

从表 2 中可知，主题数为 2 的时候跑题检测结果为平均准确率为 84.17%，平均查全率为 89.40%，平均 F 值为 86.55%。为了使跑题检测的效果达到最佳，实验中通过改变文档的主题数，从而得到主题数与 F 值的变化趋势，然后确定 LDA 建模时最佳的主题数目，最后根据选取的最佳的主题数得到实验的最终结果。

因为一篇文档会有多个主题，实验改变文档主题数 K 的值，K 的值依次选取 2、3、5、10、15、20、25、30。通过不同的主题数进行实验，选取一定的阈值后分别得到相应的跑题文档，作为实验跑题检测结果。根据之前人工标注好的打分结果进行对比分析，得到每个题目的准确率、查全率和 F 值，最后算出相应评价方法的平均值。因为 F 值综合考虑了准确率和查全率，所以实验最终用 F 值作为最后的评价指标。实验中通过选取不同的 K 值，可以得到相应的 F 值，不同主题数下的平均 F 值的结果如图 4 所示。

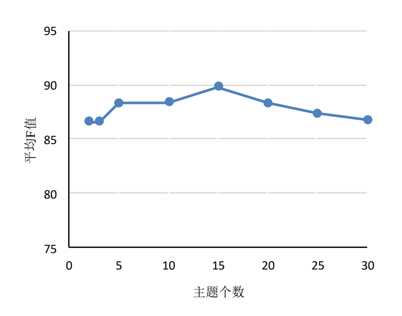


图 4　不同主题数时的平均 F 值

从图 4 可以清晰的看到平均 F 值随着不同主题数目变化的情况，发现在主题数目为 15 的时候，平均 F 值达到最高。因此本文可以确定最佳的主题数目为 15。同时在实验中发现，随着主题数目的增加实验的迭代时间也会增长。

实验中发现更改文档的主题数目 K 值时，超参数 α 的值也会随之改变。K 的值和 α 成反比的关系，显然 K 的值越大，α 的值越小，表明每篇文档包含更多的主题。对于实验中每个主题下的特征词，在文献[19]中已经证明在选取 5 个特征词的时候会取得较好效果，所以在本实验中，本文统一的对每篇文档的每个主题选取 5 个特征词进行实验。

通过本文确定的最佳主题数目进行实验，实验结果检测到的跑题文档和带有人工标注打分的跑题文档对比后，最后得到在 6 个不同题目下跑题检测的平均准确率为 91.86%，平均查全率为 88.78%，平均 F 值为 89.81%。

本文还通过基于向量空间模型的 TF-IDF 算法进行了对比实验。对比实验用同样的英文作文文档作为语料库，首先对语料库进行预处理，再利用 TF-IDF 算法把文档表示成关于词项的向量，其次待检测作文分别与给定 5 篇范文计算余弦相似度，然后根据相似度结果做均值处理作为该文档的结果，最后根据阈值筛选出对应题目的跑题文档。与本实验使用的评价方法相同，该实验最后用 F 值作为评价指标，通过 6 组实验所得到的跑题检测平均 F 值为 77.4%。

本文方法与基于向量空间模型的 TF-IDF 算法的 F 值对比结果如图 5 所示：

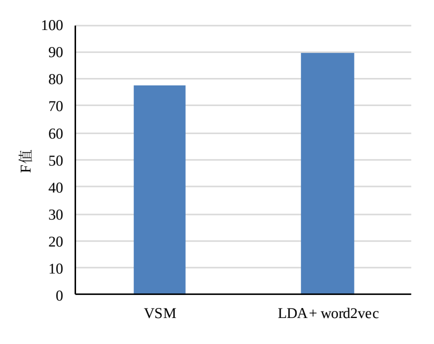


图 5　不同方法的 F 值对比

根据两个方法的实验结果对比分析，从图 5 中可以看到本文提出的算法效果更好，可以准确的分析出文档中词项的语义信息，还可以得到文档中的主题分布情况，这些因素对检测文档是否跑题很有帮助。在保证一定的准确率的情况下，本文的算法相比向量空间模型的 TF-IDF 算法，可以检测出更多的跑题作文，F 值有明显的提升，算法具有可靠性。在对比实验中发现，其中的两组实验，本文提出的方法找到了该题目下的所有跑题作文，准确率较高，而基于向量空间模型的 TF-IDF 算法并没有检测到该题目下的所有跑题作文，在未检测到的作文中，发现有 0 分作文，虽然作文的内容不是空白的，但是其主题是跑题的，本文提出的算法就可以很好地检测到这些文档。这一事实也反映了基于向量空间模型的 TF-IDF 算法的一个最大的缺点，它仅仅是通过 TF（词频）和 IDF（逆文档频率）计算，不能很有效地判断文档中词项本身的语义信息，具有一定的局限性。

本文的跑题检测算法检测到的跑题作文可以达 88%以上，准确率也比较高，同时比向量空间模型下的 TF-IDF 算法更具有效性，可以在短时间内高效的筛选出相应题目的跑题作文，这可以为教师阅卷节省了很多时间。

5 结束语

本文利用 LDA 对文档建模，可以方便地提取出文档的主题及其特征词，并用 word2vec 对其训练，训练后的结果能更准确地表达词语之间的语义，再用 LDA 和 word2vec 对文档进行主题相关度计算，实验结果表明，通过该算法有效地检测了跑题作文。本文提出的算法对英语教学包括英语竞赛的阅卷具有智能化辅助作用，该算法通过计算机可以有效地模拟教师快速、客观、公正、自动地对英文作文进行处理，并且把相应题目下的跑题作文筛选出来，减少了教师阅卷的主观因素影响，进而提高了阅卷的效率，弥补了人工无法在短时间内对大量英文作文快速有效的检测跑题方法的缺陷。

本文在用 LDA 建模确定主题数时仅用 F 值作为参考，而没有考虑更好的确定主题数的计算理论。考虑到 LDA 模型很容易扩展，下一步工作将准备在 LDA 模型的基础上，继续研究并且改进其对文档建模及主题数确定的方法。

参考文献：

[1] 陈志鹏，陈文亮，朱慕华. 利用词的分布式表示改进作文跑题检测 [J]. 中文信息学报, 2015, 29 (5): 178-184, 203.

[2] Deane P. On the relation between automated essay scoring and modern views of the writing construct [J]. Assessing Writing, 2013, 18 (1): 7-24.

[3] 翟延冬，王康平，张东娜，等. 一种基于 WordNet 的短文本语义相似性算法 [J]. 电子学报, 2012, 40 (03): 617-620.

[4] 游彬，严岳松，孙英阁，等. 基于 HowNet 的信息量计算语义相似度算法 [J]. 计算机系统应用, 2013, 22 (01): 129-133.

[5] 张志飞，苗夺谦，高灿. 基于 LDA 主题模型的短文本分类方法 [J]. 计算机应用, 2013, 33 (06): 1587-1590.

[6] 姚全珠，宋志理，彭程，等. 基于 LDA 模型的文本分类研究 [J]. 计算机工程与应用, 2011, 47 (13): 150-153.

[7] 王振振，何明，杜永萍. 基于 LDA 主题模型的文本相似度计算 [J]. 计算机科学, 2013, 40 (12): 229-232.

[8] Arora S, Ge R, Halpern Y, et al. A Practical Algorithm for Topic Modeling with Provable Guarantees [C]// Proc of International Conference on Machine Learning. 2012: 280-288.

[9] Farrahi K, Gaticaperez D. Discovering routines from large-scale human locations using probabilistic topic models [J]. ACM Trans on Intelligent Systems & Technology, 2011, 2 (1): 1-27.

[10] Link W A, Eaton M J. On thinning of chains in MCMC [J]. Methods in Ecology & Evolution, 2012, 3 (1): 112-115.

[11] 马海云. 基于 Gibbs 抽样的测试用例生成技术研究 [J]. 自动化与仪器仪表, 2011, (02): 11+14.

[12] 唐明，朱磊，邹显春. 基于 Word2Vec 的一种文档向量表示 [J]. 计算机科学, 2016, 43 (06): 214-217, 269.

[13] Pennington J, Socher R, Manning C. Glove: Global Vectors for Word Representation [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2014: 1532-1543.

[14] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space [J]. Computer Science, 2013.

[15] Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and their Compositionality [C]// Advances in Neural Information Processing Systems. 2013: 3111-3119.

[16] 王鹏，高铖，陈晓美. 基于 LDA 模型的文本聚类研究 [J]. 情报科学, 2015, 33 (01): 63-68.

[17] 胡吉明，陈果. 基于动态 LDA 主题模型的内容主题挖掘与演化 [J]. 图书情报工作, 2014, 58 (02): 138-142.

[18] 周练. Word2vec 的工作原理及应用探究 [J]. 科技情报开发与经济, 2015, 25 (2): 145-148.

[19] 吴恺，王莹. 基于提及关系的微博用户知识发现初探 [J]. 图书与情报, 2015 (2): 123-127.